Clock Event Handling

Implemented support

Wed, Jun 25, 2003

The clock event decoder is part of the Digital IndustryPack (IP) board in the IRM front end. This note describes how the support for clock events works in the IRM system, with a note about the PowerPC.

Clock event interrupt

Clock events are decoded on the Digital IP board and placed into a FIFO. Whenever the FIFO is not empty, an interrupt occurs. The interrupt routine accesses the FIFO status, and if the FIFO is not empty, it reads the event# from the FIFO and updates the corresponding entry in the Event Times table, recording the time stamp (copy of the local microsecond counter) of the interrupt and the delta time between this event and the previous occurrence of the same event. (An Event Times table entry is cleared if an event has not occurred for a half hour.)

If the `EVTLOG` data stream queue exists, it writes the event number and the low 24 bits of the time stamp into the queue. It then sets a bit in the dynamic clock event bit map array of 32 bytes (256 bits).

There is a user-selected event# that can be set via the Clock Events page application. If the current event# matches the selected event number, count the number of matches and record the time within the current cycle at which the event occurred. Then loop back to check the FIFO status again. When the status shows that the FIFO is empty, exit the loop and histogram the number of events found in the FIFO during this event interrupt routine execution. The histogram has 8 entries in it, so it can count from 0–7 events per interrupt. These diagnostic results can be viewed on the Clock Events page.

It is not unusual for two clock events to occur nearly simultaneously. Because they are encoded on a serial clock signal, the closest they can actually occur is 1.2 μs. The interrupt routine advances the time stamp by 1 for each successive event it finds in the FIFO, which is a good approximation for the case that the clock events are as close as possible. According to the Interrupt Timing diagnostics, the interrupt routine lasts at least 10 μs. Monitoring the current situation at the time of this writing, it is most common that clock events occur singly, but 25% of the time, two events are detected in one interrupt routine execution. Perhaps 5% of the time, three or four interrupts occur together.

Every cycle, early in the Update task execution, the `CLKEVTBM` routine is called to update the static bit map from what is found in the dynamic bit map that the event interrupt routine sets. Eight longwords are sampled from the dynamic bit map, exclusive-ORed into the same longword just read (to remove the bits that were just sampled) and written to the static event bit map. If another clock event occurs during this sampling operation, it may be lost this time, but it will be sampled next cycle. Most of the events of interest occur synchronously with the cyclic operation of the system, so such occurrences are infrequent. The static bit map is not altered until the next cycle. Code that cares about clock events samples the static bit map to detect whether a given event has occurred. This includes data request processing logic that has to determine when is the right cycle to update and return data in reply to an active event-based request.

The page application `EVTQ` allows capturing selected clock events by scanning the data stream queue for clock event activity. See the *Clock Events Diagnostic* note for more information.

The page application `EVTS` shows current clock event activity by monitoring the static bit map and the clock event diagnostics. See the *Clock Events Page* note for more information.

PowerPC version

The above description is for the IRM, but the PowerPC systems, which use a Digital PMC board, are set up in the same way. The low memory addresses are mapped in PPC systems so that memory data requests result in accessing the same variables and tables described below.

Memory addresses of interest

Name	*Address*	*Size*	*Function*
`CLKEVTB`	`2F80`	32	Dynamic clock event bit map
`CLKEVTB+32`	`2FA0`	32	Static clock event bit map
`EVTIMH`	`2FE0`	32	Event times table header
`EVTIME`	`3000`	2048	Event times table (256 entries)

`EVTIME` 8-byte entry structure:

`INT32`	`EVTTIM`	Time of last event (1 MHz counter)
`INT32`	`EVTDEL`	Time between last two of these events

`EVTIMH` 32-byte structure:

`INT32`	`EVTNEVT`	#event interrupts
`INT16`	`EVTNEVL`	#events last interrupt
`INT16`	`EVTNEVM`	Maximum #events per interrupt
`INT16`	`EVTINDX`	Event index# for retiring old entries
`INT16`	`EVTSELE`	Selected event#
`INT16`	`EVTSELN`	Number of selected events
`INT16`	`EVTRTCY`	Relative time in cycle for last selected event
`INT16`	`EVTHIST[8]`	Histogram of #events found per interrupt

`EVTLOG` data stream queue (size = 4K) based at `0xA000`